대규모 웹 지리정보시스템을 위한 메모리 상주 공간 데이터베이스 클러스터†

Main Memory Spatial Database Clusters for Large Scale Web Geographic Information Systems

이재동*
Jae-Dong Lee

요 약 웹을 통해 위치기반 서비스 등과 같은 다양한 지리정보 서비스를 사용하려는 사용자가 급격하게 증가하면서, 웹 지리정보시스템도 많은 다른 인터넷 정보시스템들과 같이 클러스터 기반 아키텍처로의 변화가 요구되고 있다. 즉, 사용자의 수에 상관없이 양질의 지리정보 서비스를 지속적이며 빠르게 제공하기 위해서는 비용대비 효율, 가용성과 확장성이 높은 클러스터 기반의 웹 지리정보시스템이 필요하다.

본 논문에서는 가용성과 확장성이 높은 클러스터 기반의 웹 지리정보시스템을 설계한다. 이를 위해 메모리 상주 공간 데이터베이스들을 클러스터의 각 노드로 구성하고 전체 데이터 영역 중 일부만을 복제 처리함으로써, 각 노드가 공간 질의에 대해 공간적 근접성을 이용한 캐시 역할을 수행하도록 한다. 또한, 제안된 시스템은 단순 영역 질의외에 연산 비용이 큰 공간 조인 연산을 효율적으로 처리한다. 본 논문에서는 성능평가를 통해 제안된 기법이 기존 기법에 비해 데이터 양이 많고, 클러스터의 노드 수가 증가할수록 각각 약 23%, 30%의 향상된 성능을 갖음을 보인다.

ABSTRACT With the rapid growth of the Internet geographic information services through the WWW such as a location-based service and so on, Web GISs (Geographic Information Systems) have also come to be a cluster-based architecture like most other information systems. That is, in order to guarntee high quality of geographic information service without regard to the rapid growth of the number of users, web GISs need cluster-based architecture that will be cost-effective and have high availability and scalabilty.

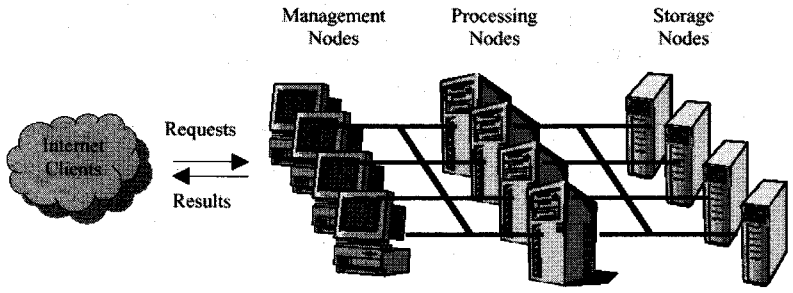
This paper proposes the design of the cluster-based web GIS with high availability and scalability. For this, each node within a cluster-based web GIS consists of main memory spatial databases which accomplish role of caching by using data declustering and the locality of spatial query. Not only simple region queries but also the proposed system processed spatial join queries effectively. Compare to the existing method, Parallel R-tree spatial join for a shared-Nothing architecture, the result of simulation experiments represents that the proposed spatial join method achieves improvement of performance respectively 23% and 30% as data quantity and nodes of cluster become large.

주요어 : 웹 지리정보시스템, 메모리 상주 공간 데이터베이스, 클러스터 시스템
Key word : Web GIS, Main-Memory Spatial Database, Cluster System

1. 서 론

최근 고성능 프로세서와 초고속 네트워크 등의 하드웨어 기술이 발전함에 따라 슈퍼컴퓨터와 같은 고가의 대형 컴퓨터를 사용하는 대신에 여러 개의 프로세싱 노드들을 클러스터링 기술을 사용하여 고속의 네트워크로 묶는 클러스터 시스템이 많이 활용되고 있다. 특히, 인터넷을 기반으로 하는 대부분의 정보 시스템들

† 이 연구는 2003학년도 단국대학교 대학 연구비의 지원으로 연구되었음.
* 단국대학교 정보컴퓨터학부 컴퓨터과학전공 부교수 letsdoit@dku.edu



〈그림 1〉 클러스터 기반의 정보 시스템

이 클러스터 기반의 하부 구조를 갖는 클러스터 시스템으로 변모하고 있다 〈그림1〉. 즉, 인터넷 환경에서 급속하게 증대되는 서비스 요구를 효과적으로 처리하고 E-Business와 관련된 응용 서비스에서 요구되는 24시간 무정지 서비스를 제공하기 위한 방안으로 여러 대의 시스템을 연결하여 논리적으로 하나의 시스템으로 운용하는 클러스터링 시스템의 필요성이 더욱 높아지고 있다.

이러한 변화의 양상은 지리정보시스템분야에도 뚜렷이 나타나는데, 인터넷 및 무선 통신 환경의 급속한 발전과 더불어 수많은 웹 및 무선 단말기 사용자들은 위치기반 서비스를 제공받기 위해 서비스 제공 서버에 접속하여 온라인 상태로 주소검색, 상호검색, 최단경로 및 실시간 교통정보 등을 이용하게 된다. 이때 사용자들의 질의가 한꺼번에 집중되면서 서버에 과도한 부하가 걸리고, 서비스 영역의 전국 확대 및 다양화에 따른 대용량 데이터 처리 부담으로 전체 시스템의 성능이 급격히 저하되거나 일시 중단되는 심각한 문제가 발생된다. 이러한 문제를 해결하기 위해서는 클러스터 기반의 웹 지리정보시스템의 설계와 클러스터 환경에 적합한 효율적인 공간 질의처리 기법이 요구된다.

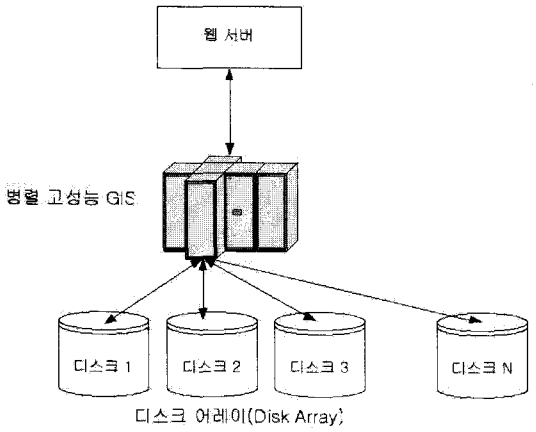
따라서, 본 논문에서는 인터넷 및 이동 컴퓨팅 환경하에서 안정되고 질 높은 지리정보 서비스를 제공하기 위해 높은 가용성과 확장성을 갖는 벡터기반 지리정보시스템 클러스터를 제안한다. 제안된 시스템은 메모리 상주 공간 데이터베이스들을 논리적인 하나의 클러스터로 구성하고 전체 데이터 영역 중 일부분만을 분할 및 복제함으로써, 각 노드가 공간 질의에 대해 공간적 친밀도를 이용한 캐시 역할을 수행하도록 한다. 또한, 본 논문은 제안된 클러스터 시스템 환경하에서 처리비용이 높은 공간 조인 연산을 효율적으로 처리하기 위한 새로운 공간 조인 기법을 제안한다. 제안된 공간 조인 기법은 병렬 공간 데이터베이스시스템과 분산 공간 데이터베이스시스템 환경에 적합토록 구현되어 있는 기존의 기법들을 개선한 것으로, 여과와 정제 연산 각각에 대한 태스크 생성 및 할당 수행에 따른 비용 문제와 최하위 두 개의 노드에서만 수행되는 한계성을 제거한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구로 다양한 지리정보시스템의 아키텍쳐와 병렬 공간 조인 기법에 대해 설명하고, 3장에서 본 논문에서 제안하는 메모리 상주 데이터베이스 클러스터 시스템의 아키텍쳐를 다룬다. 4장에서는 제안 시스템의 공간 조인 처리 과정을 설명한다. 그리고 5장에서 성능평가를 보이고, 6장에서 결론을 맺는다.

2. 관련연구

2.1 병렬 고성능 지리정보시스템

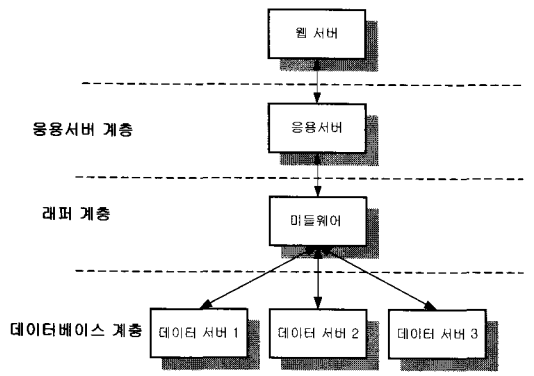
병렬 고성능 지리정보시스템은 주로 실시간 3차원 지형의 랜더링(Rendering)이나 공간분석을 통한 의사결정 등과 같은 응용분야에 사용되는 시스템으로서, 많은 분량의 지리 데이터를 빠르게 접근하여 분석 과정을 요하는 고비용의 공간 연산을 어떻게 병렬적으로 처리할 것인가에 주안점을 두고 있다. 이러한 병렬 고성능 지리정보시스템에서는 전체 데이터를 여러 개의 디스크에 어떻게 할당할 것인가 라는 디클러스터링(declustering) 문제와 전체 질의의 부하를 어떻게 여러 개의 디스크에 분산시킬 것인가 라는 부하분산(load balancing) 문제에 있어 많은 연구가 수행되어 왔다. 이러한 연구에서 사용된 시스템 환경은 〈그림 2〉와 같이 주로 SMP(Simultaneous Multithreading Processor) 시스템의 다중 디스크를 기반으로 하였다[11].



〈그림 2〉 병렬 고성능 지리정보시스템의 아키텍쳐

2.2 미들웨어 기술에 기반한 지리정보시스템

미들웨어 기술에 기반한 지리정보시스템은 주로 이질적인 구조를 갖는 다중 지리정보시스템 서버들을 접근하기 위한 방법으로 개방형 지리정보시스템 서버의 구현을 목적으로 한다. 기존의 연구로 여러 업체 및 연구소가 OpenGIS 컨소시움을 구성하여 공간데이터의 상호 운용을 위한 인터페이스 명세를 제안하고 있으며, 제안된 명세에 따라 다양한 접근 방법을 제안하고 있다[10]. 기존 연구들의 기반이 되는 시스템구조는 다음과 같다〈그림 3〉.

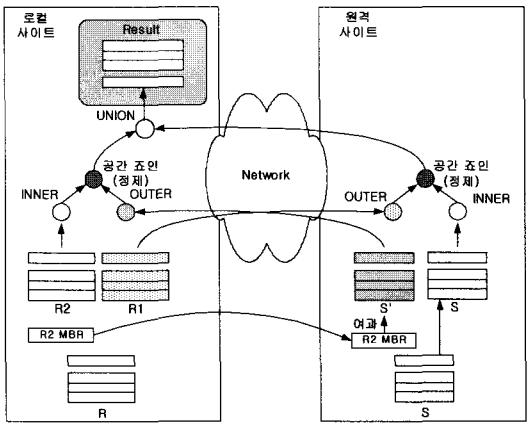


〈그림 3〉 미들웨어 기술에 기반한 지리정보시스템의 아키텍쳐

〈그림 3〉에서 데이터베이스 계층의 다양한 이질적인 서버를 접근하기 위해서 래퍼계층을 두어 각 서버별로 다르게 정의되어 있는 스키마의 매핑을 수행하게 함으로써, 클라이언트에게 하나의 공통 인터페이스를 제공하는 역할을 수행한다.

2.3 분산 공간 데이터베이스 시스템에서의 병렬 공간 조인 기법

분산 공간 데이터베이스 시스템에서 서로 다른 사이트에 존재하는 두 공간 릴레이션들 간에 질의를 수행할 경우 공간 조인 연산이 발생한다. [14]에서는 분산 공간 데이터베이스 시스템에서의 병렬 공간 조인 수행 시 사이트간 공간 조인 질의를 병렬적이며 연쇄적으로 수행하는 기법을 제안하였다. 제안된 기법은 최하위 사이트 간의 공간 조인 연산에 대해 공간 조인의 대상이 되는 영역을 균등한 수의 공간 객체가 포함되도록 대상 릴레이션을 분할한 후, 질의 수행에 참여하는 두 서버에 분배하여 병렬로 수행시켜 질의 처리를 향상시킨다. 즉, 비용이 많이 드는 두 기본 릴레이션(Base Relation)들 간의 공간 조인 연산 수행 시 입력되는 객체의 수를 반으로 줄여 공간 연산에 의한 CPU 부하와 R-tree등의 공간 인덱스 탐색 시 소요되는 디스크 I/O 비용을 줄여 전체적인 질의 수행시간을 감소시킨다.



〈그림 4〉 분산 공간 데이터베이스 시스템에서의 병렬 공간 조인 기법

즉, 〈그림 4〉에서 로컬사이트의 릴레이션 R를 R1과 R2로 분리한 후, R2와 조인 가능한 영역의 객체들의 집합인 릴레이션 S'를 원격사이트에 요청하여 R2와 S'의 공간 조인 연산을 수행한다. 이와 동시에 원격사이트에서는 S와 지역사이트로부터 전송받은 R1과의 공간 조인 연산을 수행하며, 각 사이트에서 수행한 두 개의 조인 연산의 결과를 로컬사이트에서 조합하여 클라이언트에 전송한다. 그러나, 이러한 기법은 인터넷 환경을 기반으로 하여 공간 조인 수행 비용 뿐

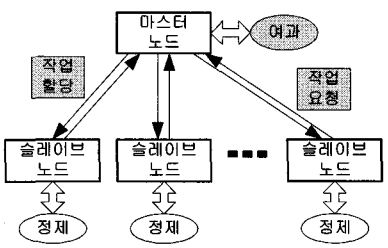
만 아니라 네트워크 상의 공간 데이터 전송 비용을 고려해야 하며, 최하위 두 사이트 노드에서만 병렬 공간 조인을 수행할 수 있다는 한계를 갖는다.

2.4 병렬 공간 데이터베이스 시스템에서의 병렬 공간 조인 기법

최근까지 병렬 공간 데이터베이스 시스템에서 고비용의 공간 조인 연산을 효율적으로 처리하기 위한 병렬 공간 조인 기법으로 다양한 연구가 진행되어 왔다[2][5][13].

Brinkhoff는 단일 할당 공간 색인 방식인 R*-tree 상에서 공유 가상 메모리(shared-virtual memory)를 가지는 병렬 시스템에서의 공간 조인을 위한 방법들을 제시하였다[2]. Hoel은 단일 할당 공간 색인을 이용한 것으로서 비겹침-정규 데이터 공간 분할 방식을 사용하는 Quad-tree에서 병렬 공간 조인 방법 제안하였다[5]. 한편, [13]에서는 데이터 집합 공간을 일정한 면적을 갖도록 X, Y 좌표축으로 등분한 후 단일 또는 다중 할당 고정 그리드(Grid)를 공간 색인으로 하는 공간 조인 기법을 제안하였다.

그러나, 위의 모든 병렬 공간 데이터베이스 시스템에서의 병렬 공간 조인 기법은 프로세서들 간의 작업 분배의 문제, 디스크 병목 현상 및 중복 결과 제거에 대한 문제들을 발생시킨다. 즉, 〈그림 5〉에서 질의를 받은 마스터 노드에서 슬레이브 노드들에게 태스크를 생성해서 각 노드별로 여과 연산을 병렬적으로 수행하여 후보 객체 쌍들로부터 중복을 제거한 후, 다시 병렬적으로 정제 연산을 수행한다. 정제 연산의 중복을 제거했지만, 여과와 정제 연산 각각에 대해 태스크 생성 및 할당 과정을 수행해야 하는 단점이 있다.



〈그림 5〉 병렬 공간 데이터베이스 시스템에서의 병렬 공간 조인 기법

3. 메모리 상주 공간 데이터베이스 클러스터의 설계

본 절에서는 논문에서 제안하는 메모리 상주 공간 데이터베이스 클러스터의 기본적인 고려사항, 아키텍처, 데이터 분할 및 복제 정책에 대해서 설명한다.

3.1 고려사항

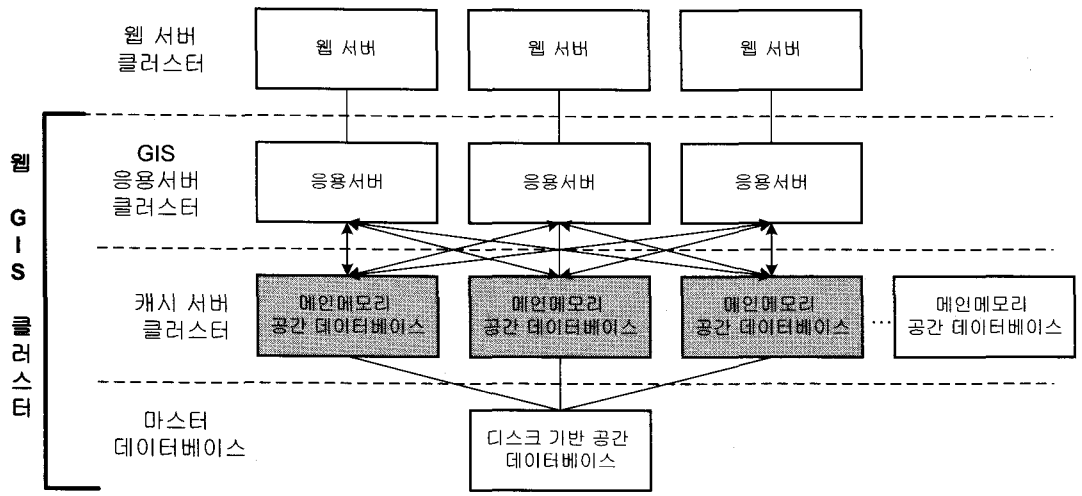
제안하는 메모리 상주 공간 데이터베이스 클러스터 시스템의 기본적인 고려사항은 다음과 같다.

- 웹 GIS 클러스터 시스템의 구축 시, 고비용의 다중 프로세서 서버 대신에 저비용의 PC 클러스터로 구성함으로써 시스템의 비용대비 효율성을 높인다.
- 대부분의 데이터베이스 응용프로그램에서와 동일하게, 웹 GIS의 클라이언트들이 요청하는 질의들은 서로 친밀도를 갖는다. 친밀도는 질의간 포함성(Containment)과 근사도(Proximity)에 기인하는데, 포함성은 공간 데이터의 경우, 행정구역별로 서로 포함관계 (예, 국가⊃도/시⊃읍(면)/구)가 성립하고, 근사도는 동일 질의에 의해 접근되는 공간데이터들이 서로 위치상으로 가까이 있기 때문이다. 이러한 특성은 데이터베이스에서 다중질의의 최적화와 고차원 색인 분야에서 이미 많은 연구가 수행되어 왔다[4][9]. 본 웹GIS클러스터에서는 이러한 질의 친밀도를 이용하기 위해, 캐시역할을 하는 각 노드의 메모리 상주 공간 데이터베이스가 전체 데이터 영역 중 일부를 분할하여 관리하도록 한다.
- 클라이언트 질의에 의한 접근 빈도가 높은 데이터 영역은 여러 노드에 복제함으로써 전체 트랜잭션 부하를 분산시킨다.
- 클라이언트가 요구하는 질의에 대하여 공간 질의의 친밀도를 기반으로 하여 해당 노드로 라우팅을 수행함으로써, 평균 트랜잭션 처리 시간을 최소로 한다.
- 질의 친밀도는 정적 또는 동적이다. 즉, 질의 패턴의 변경에 의해서 초기 친밀도가 동적으로 변동이 일어날 수 있으며, 이 경우, 동적인 로그 정보를 바탕으로 데이터 재구성을 수행함으로써 시스템의 성능을 항상 최적 상태로 유지한다.
- 기존 시스템들에서 주로 고려한 다중 영역질의의 최적화뿐만 아니라, 공간 조인질의와 같은 복합 질의에 대해서도 빠른 처리를 보장한다.

3.2 메모리 상주 공간 데이터베이스 클러스터의 아키텍처

다음의 〈그림 6〉은 웹 지리정보시스템을 클러스터

기반의 구조로 구성한 그림이다. 웹 서버와 GIS 어플리케이션 서버는 HTTP나 특정 클라이언트 프로그램의 요구를 처리하는 기능을 제공한다. GIS 어플리케이션 서버는 마스터 데이터베이스라고 불리우는 디스크 기반의 공간 데이터베이스를 통해 요구되는 지리 데이터를 저장 또는 관리한다. 메모리 상주 공간 데이터베이스 클러스터는 어플리케이션 서버와 마스터 데이터베이스의 사이에 위치하여 마스터 데이터베이스의 일부 데이터를 캐싱한다. 이러한 메모리 상주 공간 데이터베이스 클러스터의 각 노드는 캐쉬 서버라고 부르며, 클라이언트의 질의에 의해 접근되는 데이터는 미리 질의 수행전에 마스터 데이터베이스에서 캐쉬서버로 가져온다. 클라이언트 질의 중 갱신 질의는 제외하므로, 캐쉬서버와 마스터 데이터베이스간의 데이터 일관성 문제는 논외로 한다. 클라이언트가 요구하는 질의가 특정 노드에 집중되거나 특정 노드가 잘못된 연산으로 오류가 발생하는 경우, 온라인 상태에서 새로운 캐쉬서버를 클러스터에 추가하는 온라인 확장(Online Scaling)을 수행한다.

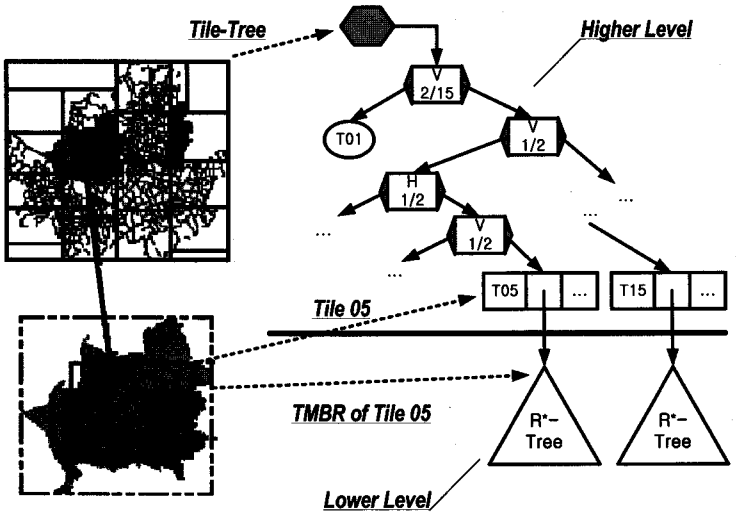


〈그림 6〉 메모리 상주 공간 데이터베이스 클러스터 시스템의 아키텍쳐

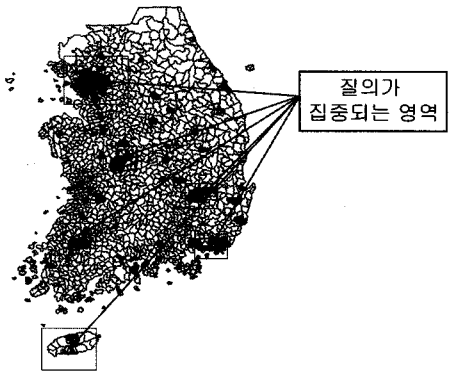
3.3 메모리 상주 공간 데이터베이스 클러스터에서의 데이터 분할 및 복제

웹 지리정보시스템에서 처리되는 클라이언트의 질의들은 서로 친밀도(Affinity)를 갖는다. 전체 공간질의들을 몇 개의 그룹으로 분류할 수 있는데, 동일한 그룹의 질의들은 동일하게 또는 부분적으로 겹치는 데이터를 접근하는 특성을 갖고 다른 그룹에 속한 질의는 서로 다른 데이터를 접근한다. 본 시스템에서는 이러한 친밀도를 갖는 공간 질의들이 동일한 노드에서 처리되게 함으로써 전체 시스템의 처리량을 향상시킨다. 이를 위한 클러스터의 각 노드에 데이터 분할을 수행한다. 이를 위해, 본 시스템에서는 전체 서비스 영역에 대한 전역 색인을 구성하기 위해 타일 기반의 공간 색인를 구성한다[12]. 제안된 공간 색인〈그림 7〉은 두 개의 레벨로 구성되는데, higher level은 데이터가 분할되는 모든 노드에 중복 저장되는 부분이고, lower level은 각 노드에 저장되는 최소 단위로써 공간 인접성이 높은 타일들이 동일한 노드에 저장되도록 데이터를 분할하게 된다[7].

또한, 〈그림 8〉과 같이 전체 데이터 중에서 대도시와 같은 인구밀도가 높은 지역은 다른 지역에 비해 월등히 많은 접근이 이루어지는 핫스팟(hot spot) 부분이라고 할 수있다. 이러한 영역은 여러 노드에 중복시킴으로써 부하를 분산시킨다. 즉, 작업 부하가 많은 부분을 중복 저장하여, 모든 노드의 메모리로부터 최대의 질의를 처리하게 하고, 포워딩 오버헤드를 최소로 한다.



〈그림 7〉 타일 기반 공간 색인



〈그림 8〉 질의 영역의 집중성

4. 메모리 상주 공간 데이터베이스 클러스터에서 공간 조인 연산의 구현

점 질의나 영역 질의와 같이 단일 공간 테이블에 연관된 사용자 질의는 수행 비용이 크지 않으므로 질의를 요구받은 노드나 다른 노드로부터 질의 처리를 요청 받은 노드에서 단독으로 처리한다. 그러나 고비용의 수행 비용이 요구되는 공간 조인 연산을 단일 노드에서 처리할 경우 해당 노드에 부하가 집중되어 전체 클러스터 시스템의 성능을 저하시키는 요인이 된다[1][3][6]. 또한, 이러한 고비용의 공간 조인 연산을 모든 노드에서 병렬로 처리하고자 중복되지 않은 공간 테이블들을 모든 노드에 전송한다면 대용량의 공간 테이블들의 전송으로 인한 네트워크 부하와 저장 공간의 낭비를 발생시킨다. 따라서 본 기법에서는 이러한 비공유 공간 데이터베이스 클러스터의 특성을 고려하여 가용성과 고성능을 위해 공간 조인 연산의 대상 테이블들을 중복하여 저장하고 있는 클러스터 노드들만 병렬 공간 조인 연산에 참여하도록 한다.

먼저, 공간 조인 질의를 수행하는 전체적인 알고리즘은 다음과 같다.

Step 1: 공간 조인 질의 요청
클러스터의 한 노드가 사용자로부터 공간 조인 질의를 요청 받는다.

Step 2: 노드 선정
공간 조인 질의를 요청받은 노드는 전역 메타 정보에서 관리되는 지역 테이블의 분할 정보를 통해, 공간 조인 질의를 병렬로 수행할 클러스터 노드들의 리스트를 구한다.

Step 3: 서브질의 전송
위에서 선택된 노드들은 해당 테이블들을 이미 중복하여 저장하고 있으므로, 공간 조인 질의를 요청받은 노드는 별도의 데이터 전송 없이 파스 트리(Parse Tree) 형식의 질의문과 자신의 노드 ID를 선택된 노드들에게 전송하여 공간 조인 연산을 요청한다.

Step 4: 각 노드별 지역 공간 조인 수행

공간 조인 연산 수행을 요구 받은 각 클러스터 노드들은 자신의 지역 메타 정보에 있는 조인 대상 테이블들의 논리적 분할 영역(Logical Partition Area)을 참조하여 자신이 수행할 공간 조인의 영역을 알아낸 후, 해당 영역 내의 공간 객체들에 대해서만 공간 조인 연산을 수행한다.

4.1 타일 공간 색인에 기반한 여과 알고리즘을 통해 여과 단계를 수행한다.

4.2 Plan-sweep 알고리즘을 이용한 정제 단계를 수행한다.

Step 5: 질의결과 전송

각 노드에서 공간 조인 연산의 결과가 만들어질 때 마다, 파이프 라인(Pipelined) 방식으로 공간 조인 질의 수행을 요청받은 노드에게 전송하며, 전송된 결과는 별도의 중복 제거 과정없이 클라이언트에게 전송한다.

4.1 공간 조인 질의의 대상 노드 선정

공간 조인 질의는 특정 지역 테이블 내에 있는 레이어들 간의 포함, 겹침, 교차 등의 관계에 있는 객체들을 찾는 것이며, 이러한 공간 조인 질의는 클러스터 내 임의의 노드에 전달된다. 공간 조인 질의를 처리해야 하는 클러스터 노드는 먼저 전역 메타 정보에서 관리되는 지역 테이블의 위치 정보를 통해 공간 조인 질의를 병렬로 수행할 클러스터 노드들의 리스트를 얻어온다. 이러한 노드들은 해당 테이블들을 이미 중복하여 저장하고 있으므로 별도의 데이터 전송 없이 파스 트리(Parse Tree) 형식의 질의문과 자신의 노드 ID를 각각의 노드들에게 전송하여 공간 조인 연산을 요청한다.

공간 조인 연산 수행을 요구 받은 각 클러스터 노드들은 자신의 지역 메타 정보에 있는 조인 대상 테이블들의 논리적 분할 영역을 참조하여 자신이 수행할 공간 조인의 영역을 알아낸 후 해당 영역 내의 공간 객체들에 대해서만 공간 조인 연산을 수행한다. 예를 들어, 클러스터 내의 노드1, 노드2, 노드3에 중복 저장된 공간 테이블 R과 S를 다음과 같은 조건으로 조인을 한다고 가정하자.

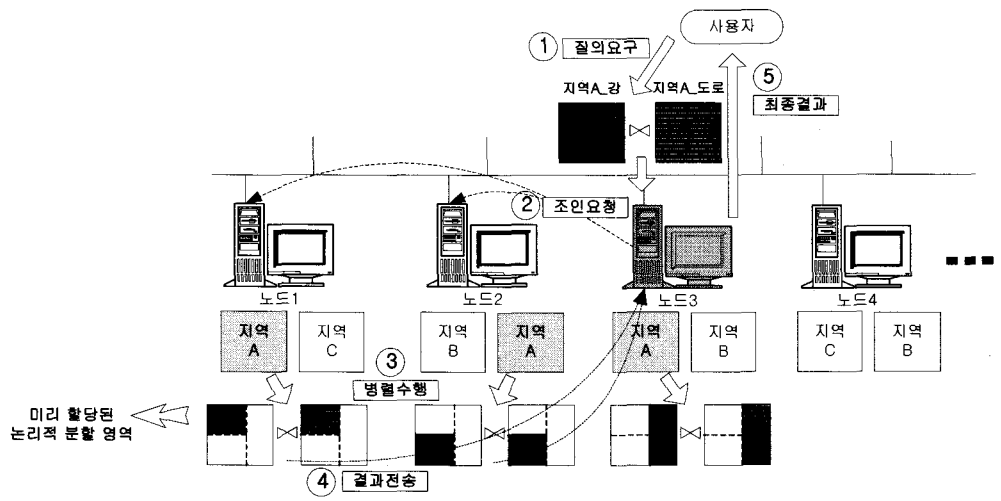
SELECT * FROM R, S WHERE CONTAINS (R.Obj, S.Obj)

위의 조인 질의문은 각각의 노드1, 노드2, 노드3에 설정된 논리적 분할 영역 LPA_Node1, LPA_Node2, LPA_ Node3에 의해 다음과 같은 3개의 서브 질의문으로 분리된다.

① SELECT * FROM R, S
WHERE INTERSECTS (LPA_Node1, R.OBJ) AND CONTAINS (R.OBJ, S.OBJ)

② SELECT * FROM R, S
WHERE INTERSECTS (LPA_Node2, R.OBJ) AND CONTAINS (R.OBJ, S.OBJ)

③ SELECT * FROM R, S



〈그림 9〉 논리적 분할 영역을 이용한 병렬 공간 조인 수행의 예

WHERE INTERSECTS (LPA_Node3, R.OBJ) AND CONTAINS (R.OBJ, S.OBJ)

공간 조인 연산 수행을 요청 받은 각 클러스터 노드는 자신의 논리적 분할 영역에 겹쳐지는 공간 객체들에 대해서만 공간 조인 연산을 수행한 후 결과 레코드들을 공간 조인 연산을 요청한 노드에게 전송하게 된다. 〈그림 9〉은 지역A 테이블 내에 있는 강과 도로 레이어에 대한 공간 조인 질의에 대해 논리적 분할 영역을 이용한 병렬 공간 조인 수행의 예를 보여주고 있다.

위의 〈그림 9〉와 같이 사용자로부터 고비용의 공간 조인 질의를 요구 받은 질의 처리 노드(①)는 전역 메타 정보를 참조하여 조인 대상 테이블들을 중복하여 저장하고 있는 노드들을 검색한 후, 해당 노드들에게 사용자 질의를 전송하여 병렬로 공간 조인 연산을 수행할 것을 요청한다(②). 공간 조인 연산 수행을 요청 받은 노드들은 자신의 지역 메타 정보에 설정된 논리적 분할 영역에 포함되는 공간 객체들에 대해서만 공간 조인 연산을 수행하게 된다(③). 각각의 노드들에서 병렬적으로 수행된 공간 조인의 결과들은 질의 처리 노드에게 전달되며(④), 질의 처리 노드는 최종 결과들을 사용자에게 전송하게 된다(⑤).

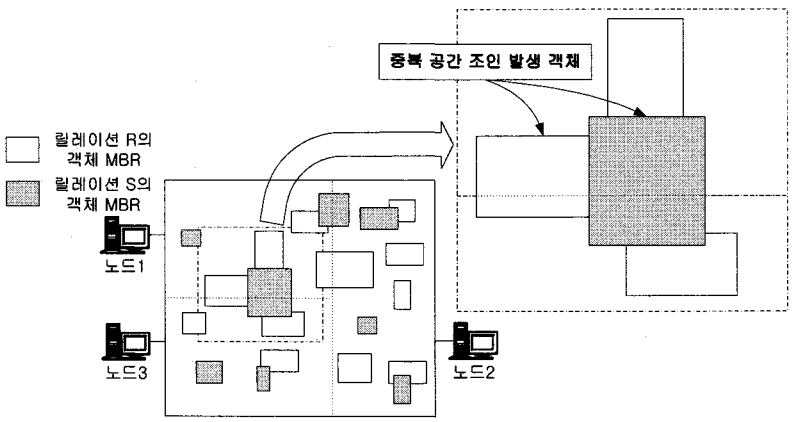
4.2 각 노드에서의 여과 단계 수행

지역 노드에서 수행되는 공간 조인 연산은 대부분의 공간 데이터베이스 시스템에서 효율적인 공간 질의 처리를 위하여 이용하는 다단계 질의 처리 방법인 여과(Filtering) 단계와 정제(Refinement) 단계로 나누어 수행된다. 본 기법에서는 객체최소경계사각형(MBR : Minimum Bounding Rectangle)의 중점을 이용한 여과 단계를 수행하여 분할 영역 경계 상에 중첩되는 공간 객체들에 대해 고비용의 정제 연산이 중복 수행되는 것을 방지한다.

공간 조인 연산 수행을 요청 받은 지역 노드는 자신의 지역 메타 정보에 있는 논리적 분할 영역에 겹쳐지는 객체들에 대해 공간 조인 연산을 수행한다. 즉, 앞에서 설명한 바와 같이 자신이 가지고 있는 논리적 분할 영역에 INTERSECT 되는 공간 객체들에 대해 여과 및 정제 연산을 수행하게 된다. 그러나 논리적 분할 영역과 같이, 공간 분할을 기반으로 병렬 공간 조인 연산을 수행할 경우에는 〈그림 10〉과 같이 분할 영역 경계선 상에 존재하는 객체들에 대해 고비용의 공간 조인 연산이 중복 수행되는 문제가 발생한다.

이러한 문제를 해결하기 위해 기존의 병렬 공간 데이터베이스 시스템에서는 병렬 공간 조인 수행 시 조인 수행 연산 전반에 걸친 작업을 제어하는 마스터 노드가 여과 연산을 여러 슬레이브 노드들에게 분배하여 병렬로 수행하도록 한 다음, 여과 연산의 결과인 후보 객체 쌍들을 전송 받아 이들 중 중복되는 것들을 제거한 후에 다시 슬레이브 노드들에게 할당하여 병렬로 정제 연산을 수행하도록 하는 방법을 사용한다. 만약 메모리 상주 구조의 클러스터와 같이 독립적인 여러 노드에서 병렬로 수행할 경우에는 고비용의 공간 조인 연산을 중복적으로 수행한 후에 최종 결과들을 병합하는 단계에서 중복 결과들을 제거하는 것이 일반적이다.

이러한 중복 조인 연산과 중복 결과 제거 등의 비용을 없애기 위해 본 논문에서는 분할 영역의 경계선 상에 존재하는 공간 객체들에 대한 고비용의 중복 조인 연산을 제거하기 위해 공간 객체들의 MBR 중점의 위



〈그림 10〉 중복 공간 조인이 발생하는 객체의 예

치를 이용한 여과 기법을 제안한다.

〈그림 11〉은 4개의 클러스터 노드들에 중복 저장되어 있는 공간 테이블 A와 B에 대한 공간 조인 연산을 나타낸 예이다. 공간 객체 A1과 B1은 논리적 분할 영역 NODE1_LPA과 NODE2_LPA의 경계 선상에 존재하며 기존의 병렬 공간 조인 기법에서는 A1과 B1의 공간 조인 연산을 노드1과 노드2에서 중복하여 수행하는 문제점이 있다. 또한 위의 예에서 중복 공간 조인 연산의 문제를 해결하기 위해 분할 영역 경계선 상에 존재하는 객체 A1, B1, B2에 대해 논리적 분할 영역 NODE1_LPA을 관리하는 노드1에서만 처리할 경우 논리적 분할 영역 NODE2_LPA내에 있는 공간 객체 A2가 조인 연산에서 누락되는 경우가 발생한다.

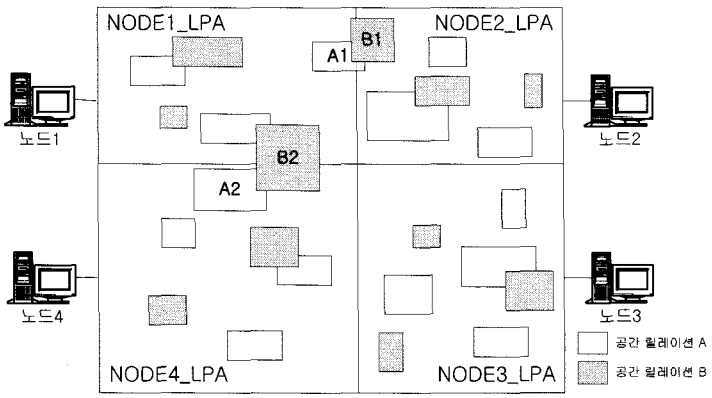
본 기법에서는 논리적 분할 영역의 경계선 상에 존재하는 공간 객체들에 대한 공간 조인 연산 중 여과 단계에서 기준 테이블을 설정한 후 기준 테이블과 후보 객체들의 MBR 중점의 위치에 따라 선별적으로 최종 후보 객체 쌍을 출력하여 중복되는 후보 객체 쌍들로 인한 고비용의 중복 정제 연산이 발생하지 않도록 한다.

논리적 분할 영역 경계선 상에 존재하는 서로 다른 공간 테이블 내의 두 공간 객체에 대한 여과 연산 시 기준 테이블과 객체 MBR의 중점의 위치에 따라 후보 객체 쌍으로 출력될 수 있는 경우는 다음과 같다.

① 두 공간 객체의 MBR 중점이 논리적 분할 영역 내에 있는 경우

② 기준 테이블 내의 공간 객체의 MBR 중점만 논리적 분할 영역 내에 있는 경우

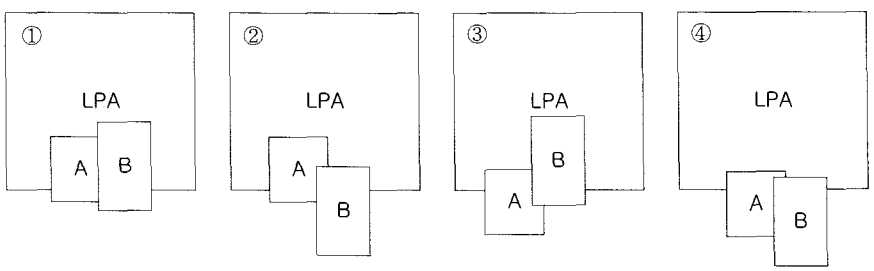
③ 기준 테이블 내의 공간 객체의 MBR 중점만 논리적 분할 영역 외에 있는 경우



〈그림 11〉 클러스터 노드 상에서 공간 테이블 A와 B의 공간 조인 연산 예

④ 두 공간 객체의 MBR 중점이 논리적 분할 영역 외에 있는 경우

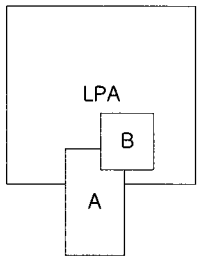
이 경우를 도식화하면 〈그림 12〉와 같다. 여기서



〈그림 12〉 논리적 분할 영역 경계선 상에서 후보 객체 쌍이 될 수 있는 경우

LPA는 논리적 분할 영역이고 A는 기준 테이블 내에 있는 공간 객체의 MBR이다.

본 기법에서는 이 경우들 중 기준 테이블의 공간 객체의 MBR 중점이 자신의 논리적 분할 영역 내에 있는 〈그림 12〉의 ①과 ②의 경우에 대해서만 최종 후보 객체 쌍으로 출력하여 정제 연산을 수행하도록 하고 ③과 ④의 경우는 해당 영역 하단에 존재하는 논리적 분할 영역을 관리하는 노드에서 수행하게 된다. 그러나, 해당 노드에서 이들 경우에 대해서만 공간 조인을 수행하게 되면 아래의 〈그림 13〉과 같이 다른 논리적 분할 영역 내에 완전히 포함되는 공간 객체들이 여과 수행에서 누락되는 경우가 발생할 수 있다.



〈그림 13〉 논리적 분할 영역을 이용한 여과 연산에서 누락될 수 있는 경우

그러므로, 본 기법의 여과 과정에서 최종 후보 객체 쌍으로 산출될 수 있는 경우는 다음과 같다.

① 두 공간 객체의 MBR 중점이 논리적 분할 영역 내에 있는 경우

② 기준 테이블 내의 공간 객체의 MBR 중점만 논리적 분할 영역 내에 있는 경우

③ 기준 테이블 외의 공간 객체의 MBR이 논리적 분할 영역에 완전히 포함되는 경우

〈그림 14〉는 LPA의 논리적 분할 영역을 갖는 클러스터 노드에서 최종 후보 객체 쌍으로 출력할 수 있는 경우와 조건식을 도식화 한 것이다. 여기서 여과 연산은 모든 공간 테이블들에 효율적인 공간 연산을 위해 모든 공간 테이블마다 구축된 타일 공간 색인을 이용한다. 타일 공간 색인을 이용할 경우 탐색 영역 제한 기법(Restricting the search space), 공간 정렬 및 평면 스위핑 기법(Spatial sorting and plane sweep) 등을 이용하여 더욱 효과적으로 여과 연산을 수행할 수 있다. 이러한 중복 후보 객체 제거 기법을 적용한 타일 공간 색인 기반의 여과 알고리즘은 다음과 같다.

〔ALGORITHM 1. Tile_Spatial_Index_Based_Filtering_Algorithm〕

Input:

R, S: nodes of tile-based spatial index

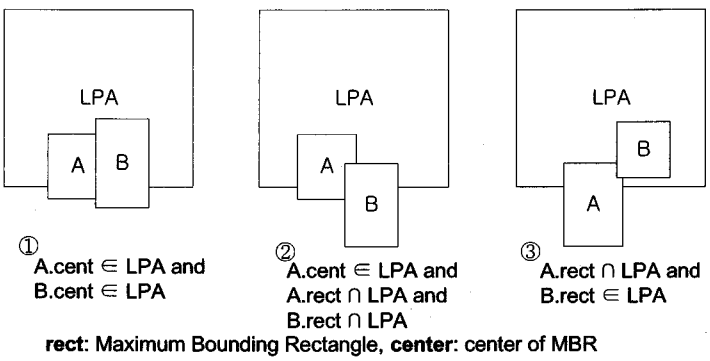
LPA: Logical partition area

Output:

Result sets of objects

SpatialJoinFiltering (R, S, LPA)

```
01  FOR (all E_S ∈ S) DO
02    FOR (all E_R ∈ R with E_R.rect
          ∩ E_S.rect ≠ Ø) DO
03      IF(R is a leaf page) THEN
04        IF(E_R.cent ∈ LPA and E_S.cent ∈
            LPA) THEN
05          Output (E_R, E_S)
06        ELSE IF(R_R.cent∈LPA and E_R.rect
            ∩LPA and E_S.rect∩LPA) THEN
```



〈그림 14〉 논리적 분할 영역 경계 상의 공간 객체가 후보 객체가 되는 경우

```
07              Output (ER, ES)
08        ELSE IF(ES.rect ∈ LPA and ER.rect
                    ∩ LPA) THEN
09              Output (ER, ES)
10          END IF
11        ELSE
12          ReadPage (ER.ref)
13          ReadPage (ES.ref)
14        SpatialJoinFiltering(ER.ref, ES.ref)
15        END IF
16      END FOR
17    END FOR
END OF ALGORITHM 1
```

위 알고리즘의 라인 번호 4에서 10은 〈그림 14〉에서 설명된 바와 같이 논리적 분할 영역 경계 상의 공간 객체가 후보 객체가 되는 세가지 경우에 대해서 각각 여과 단계를 수행한다.

4.3 정제 단계 수행 및 공간 조인 결과의 전송

여과 연산을 수행한 결과로 출력되는 후보 객체 쌍에 대해 지역 노드에서는 후보 객체로 선정된 두 공간 객체가 실제로 공간 조인 조건을 만족하는지를 여부를 확인하기 위한 고비용의 정제 연산을 수행한다. 이러한 정제 연산은 후보 객체 쌍으로 선정된 공간 객체들을 구성하고 있는 모든 점들에 대해 기하학적 연산을 통해 교차 여부를 알아내며 본 논문에서는 교차 연산 수행을 위해 가장 보편적으로 사용되고 있는 Plan-sweep 알고리즘[11]을 사용한다. Plan-sweep 알고리즘은 전처리 단계와 라인 교차 여부 검사 단계로 구성된다. 전처리 단계에서는 공간 객체를 다각형의 정보로 만들고 모든 꼭지점들을 X축을 기준으로 정렬한다. 라인 교차 여부 검사 단계에서는 수직선(Sweep-line)이 왼쪽에서 우측으로 지나가면서 만나는 꼭지점들에 대해 수직선과 교차하는 간선(edge)들을 동적인 자료 구조에 저장한다. 여기서 만약 서로 다른 다각형의 간선들이 교차되는 것을 찾아낸다면 수행을 마치고 결과를 반환하게 된다. 각 지역 노드에서는 이러한 정제 연산을 통해 조인 조건을 만족하는 결과 레코드들이 나올 때마다 파이프 라인(Pipelined) 방식으로 공간 조인 질의 수행을 요청한 질의처리 노드에게 전송하며 질의처리 노드에서는 전달받은 결과 레코드들에 대해 별도의 중복 제거 과정없이 곧바로 사용자에게 전송하게 된다. 이러한 질의 처리 노드에서의 알고리즘은 다음과 같다.

[ALGORITHM 2. Spatial_Join_Algorithm]

```
Input:
  ParseTree: Spatial Join Parse Tree
  RelationID: Region Relation ID
Output:
  Result of Spatial Join: TRUE or FALSE
BOOL ParallelSpatialJoin (ParseTree,
RelationID)
01 NodeID: Current Node ID
02 ClusterNodes[]: Cluster Nodes List
03 NodeCount: Integer Variable
04 ResultRecords: Query Result Records
05 ClusterNodes ← GetClusterNodesFrom
                      Meta( RelationID )
06 NodeCount ← ClusterNodes.Count

07 FOR (NodeCount 〉 0) DO
08    ForwardQuery (ParseTree, Cluster
      Nodes[nCount], NodeID)
09    NodeCount ← NodeCount - 1
10 END FOR
11 CreateThread (RecieveResult, Cluster
   Nodes)
12 WHILE (1)
13    ResultRecords = LocalSpatialJoin
      (ParseTree)
14    IF ResultRecords 〈〉 NULL THEN
15      SendResultToUser (ResultRecords)
16    ELSE
17        bRet ← TRUE
18      break
19    END IF
20 END WHILE
21 return bRet
END OF ALGORITHM 2
```

위 알고리즘에서 라인번호 07에서 10은 공간 조인 질의에 참여하는 노드들에게 서브 질의를 전송하는 부분이고, 라인번호 13에서는 각 지역 노드에서 수행된 결과를 파이프라인 방식으로 구하고, 라인번호 15에서는 더 이상의 결과 레코드가 없을 때 까지 사용자에

게 전송한다.

5. 성능평가

본 장에서는 기존의 비공유 구조의 병렬 공간 데이터베이스 시스템에서 사용되고 있는 병렬 R-tree 공간 조인 기법과 본 논문에서 제안한 메모리 상주 데이터베이스 클러스터의 논리적 분할 영역을 이용한 병렬 공간 조인 기법을 처리 속도를 기준으로 비교 평가한다.

5.1 평가 환경

본 평가에서 비교할 논리적 분할 영역을 이용한 병렬 공간 조인 기법과 기존의 병렬 공간 조인 기법은 비공유 기반의 메모리 상주 공간 데이터베이스 클러스터 시스템에서 구현되었다. 공간 데이터와 비공간 데이터를 통합 관리할 수 있는 공간 데이터베이스 관리 시스템을 하나의 구성 노드로 하는 비공유 공간 데이터베이스 클러스터 시스템이며 웹 GIS 시스템의 후위 시스템으로 사용되어 사용자 질의에 대해 빠르게 처리할 수 있는 고성능과 다수의 사용자로부터의 대규모 질의들을 처리할 수 있는 고가용성 및 새로운 클러스터 노드의 추가가 간단한 높은 확장성을 제공한다.

본 기법의 평가 대상이 되는 병렬 공간 조인 기법은 관련 연구 [8]에서의 비공유 구조의 병렬 R-tree 공간 조인 기법(Parallel R-tree Spatial Join for a Shared-Nothing Architecture)이다. 해당 기법은 비공유 구조의 병렬 공간 데이터베이스 시스템에서 사용되고 있는 마스터/슬레이브 노드 구조의 병렬 공간 조인 기법으로써 질의 처리 노드를 마스터 노드로 하고 그 외의 노드들을 슬레이브 노드로 하는 R-tree 공간 색인 기반의 병렬 공간 조인 기법이다.

고비용의 공간 질의를 효율적으로 처리하기 위해서는 공간 색인이 반드시 필요하므로 본 장에서의 평가에 사용되는 모든 릴레이션에 타일 공간 색인을 구축하였다. 또한 성능 평가 시 질의 처리 노드에서 해당 결과들을 사용자에게 전송하는 시간은 모든 기법에서 동일하므로 성능 평가 결과에서 제외시켰다. 평가에 사용된 시스템의 하드웨어 및 소프트웨어 환경은 모든 노드에서 동일하며, 자세한 사항은 아래의 〈표 1〉과 같다. 본 실험에서는 성능평가 모형을 구성하였고, 모형 시스템은 실제 구현 시스템에서 갖는 노드 수의 제한을 받지 않도록 CSIM 라이브러리를 이용하여 구현되었다[15][16].

〈표 1〉 성능 평가에 사용된 시스템 환경

구분	내역
기종	IBM PC
운영체제	Windows 2000 Professional
개발 언어	Visual C++ 6.0, CSIM 1.9
메인 메모리	768MB
노드간 데이터 전송 속도	1Gbps
네트워크 지연 시간	0.1ms
중앙처리장치 속도	1.5Ghz
디스크 접근 속도	10ms
레코드 접근 속도	0.2ms
버퍼 적중률	80%
쓰레드 수	50
하드디스크	40GB

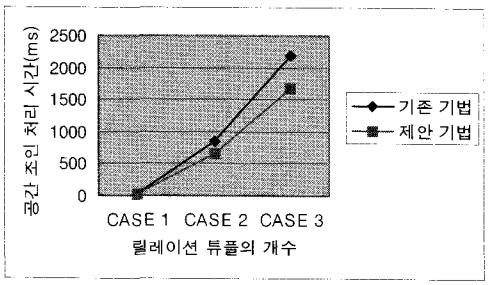
5.2 릴레이션 크기에 따른 병렬 공간 조인 성능의 평가

본 평가에서는 기존의 병렬 공간 조인 기법과 제안 기법을 공간 조인 대상 릴레이션들의 크기를 증가시키며 각각의 경우에 대한 질의 처리 시간을 비교하였다. 평가에 사용된 공간 릴레이션들에 대한 자세한 정보는 〈표 2〉와 같다. 공간 조인 수행은 서로 겹쳐지는 객체들을 결과로 하는 INTERSECT 연산을 이용하였다. 병렬 공간 조인 연산에 참여하는 클러스터 노드의 수는 4대로 하였다.

〈표 2〉 평가 릴레이션들의 상세 정보

	대상 릴레이션명	릴레이션의 크기(Kbytes)	튜플의 총 개수
CASE 1	지하철	42	112
	행정동계	80	100
CASE 2	강	498	1,220
	동경계	607	587
CASE 3	건물	1,348	7,739
	도로	1,394	6,482

〈그림 15〉의 결과를 보면 릴레이션의 크기가 커짐에 따라 기존 기법보다 제안 기법이 더 빠른 시간 내에 질의 처리를 수행하는 것을 알 수 있다.

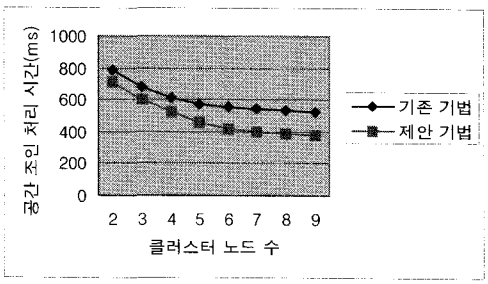


〈그림 15〉 릴레이션 크기에 따른 병렬 공간 조인 성능의 변화

이것은 마스터 노드에서 타일 공간 색인의 중간 노드(Internal node) 상에서 부작업들을 생성하여 공간 데이터의 지역적 인접성을 고려하지 않고 슬레이브 노드들에게 할당하여 병렬적으로 수행하는 기존 기법에 비해 작업 생성 등의 과정없이 이미 각 노드에 할당된 논리적 분할 영역을 이용하여 모든 노드에서 병렬적으로 공간 조인 연산을 수행하는 제안 기법이 더 효율적임을 알 수 있다. 제안된 기법은 기존기법에 비해 CASE 1, CASE 2, CASE 3에 대해서 약 22%, 23%, 23%의 향상된 결과를 얻었다.

5.3 클러스터 노드 수에 따른 병렬 공간 조인 성능의 평가

본 평가에서는 공간 조인 연산을 클러스터 노드의 수를 증가시키며 기존 기법과 제안 기법의 질의 처리 시간을 비교하였다. 평가에 사용된 공간 릴레이션은 CASE 2의 강과 동경계 릴레이션이며 공간 조인 연산은 INTERSECT 이다.



〈그림 16〉 클러스터 노드 수에 따른 병렬 공간 조인 성능의 변화

위의 〈그림 16〉의 결과를 보면 클러스터 노드의 수가 증가함에 제안 기법의 수행시간이 더 큰 폭으로 감소하는 것을 알 수 있다. 즉, 클러스터의 노드 수가 2, 3, 4, 5, 6, 7, 8, 9일때, 각각 약 9%, 12%, 15%, 21%, 25%, 27%, 28%, 29%, 30% 의 향상된 결과를 얻었다. 이것은 클러스터 노드가 늘어남에 따라 마스터 노드와 슬레이브 노드간의 메시지 전송 수의 증가와 공간 데이터의 지역적 인접성을 고려하지 않은 부작업들의 할당으로 인해 서로 다른 노드에서 동일한 공간 객체에 대한 디스크 I/O 횟수가 증가하게 되는 기존 기법에 비해 제안 기법은 노드간 메시지의 전송이 필요 없으며 공간 데이터의 지역접 인접성을 고려한 MBR 기반의 논리적 분할 영역을 사용하므로 디스크 I/O 횟수가 동일하기 때문으로 볼 수 있다.

6. 결론

인터넷 및 무선 통신 환경의 급속한 발전과 더불어 수많은 웹 및 무선 단말기 사용자들은 위치기반 서비스를 제공받기 위해 서비스 제공 서버에 접속하여 온라인 상태로 주소검색, 상호검색, 최단경로 및 실시간 교통정보 등을 이용하게 된다. 이때 사용자들의 질의가 한꺼번에 집중되면서 서버에 과도한 부하가 걸리고, 서비스 영역의 전국 확대 및 다양화에 따른 대용량 데이터 처리 부담으로 전체 시스템의 성능이 급격히 저하되거나 심지어 일시 중단되는 심각한 문제가 발생된다. 더욱이, 이러한 문제는 서비스 제공자의 신뢰성에 큰 영향을 미치고 질 높은 서비스를 원하는 다양한 사용자들의 욕구를 충족시키지 못하는 결과를 초래한다. 또한, 서비스의 다양화로 많은 질의가 고비용의 연산 비용이 요구되는 공간 조인 질의를 필요로 하고 있다.

본 논문에서는 인터넷 및 이동 컴퓨팅 환경 하에서 안정되고 질 높은 지리정보 서비스를 제공하기 위해 높은 가용성(Available)과 확장성(Scalability)을 갖는 벡터기반 지리정보시스템 클러스터를 제안했다. 제안된 시스템은 메모리 상주 공간 데이터베이스들을 논리적인 하나의 클러스터로 구성하고 전체 데이터 영역 중 일부분만을 분할 및 복제 처리함으로써, 각 노드가 공간 질의에 대해 공간적 근접성을 이용한 캐시 역할을 수행하도록 하였다. 또한, 고비용의 공간 조인 연산을 병렬적으로 수행하는 기법을 제안하였다. 제안된 기법은 조인에 참여하는 노드마다 자신에게 할당된 논리적 분할 영역 내의 공간 객체들에 대해서만 공간 조인을 수행하도록 함으로써 기존의 병렬 공간 데이터베이스 시스템에서 병렬 공간 조인 수행 시 필요로 했던 각 노드간 작업 생성 및 할당 단계와 노드간 메시지

전송 등을 제거하여 고비용의 공간 조인 질의를 빠르게 처리하였다. 또한, 공간 조인 연산의 여과 단계에서 논리적 분할 영역 경계 상의 공간 객체들에 대해 객체 MBR 중점의 위치에 따른 여과 기법을 사용하여 해당 객체들에 대해 고비용이 요구되는 정제 연산을 여러 노드에서 중복 수행하는 것을 방지하였으며 중복 결과 등을 제거하기 위한 병합(Union) 과정을 수행하지 않으므로 빠른 사용자 응답시간을 만족시켰다. 본 논문에서는 실험 모형에 의한 성능평가를 통해 제안된 기법이 기존 기법들에 비해 데이터 양이 많고, 클러스터의 노드 수가 증가할수록 각각 약 23%, 30%의 향상된 성능을 갖음을 보였다.

참고문헌

[1] Brinkhoff, T., Kriegel H.P., Schneider, R. and Seeger B., "Multi-Step Processing of Spatial Joins", Proc. ACM SIGMOD Int. Conf. On Management of Data, 1994, pp.197-208.

[2] Brinkhoff, T., Kriegel, H.P. and Seeger,B. "Parallel Processing of Spatial Joins Using R-trees", Proc. of 12th Intl. Conf. on Data Engineering (ICDE'96), 1996.

[3] Brinkhoff, T., Kriegel H.P., and Seeger B. : "Efficient Processing of Spatial Joins Using R-trees", Proc. ACM SIGMOD Int. Conf. On Management of Data, 1993, pp.237-246.

[4] Chung-Min Chen and Rakesh K. Sinha, "Analysis and Comparison of Declustering Schemes for Interactive Navigation Queries", IEEE Transactions on Knowledge and Data Engineering, Vol. 12, No. 5, 2000, pp.763-778.

[5] Hoel E. G., and Samet, H. "Data-Parallel Spatial Join Algorithms", Proc. of Int. Conf. on Parallel Processing, 1994, pp. 227-234.

[6] Tamura K., Nakano, Y. Kaneko, K. Makinouchi A., "The Parallel Processing of Spatial Selection for Very Large Geo-Spatial Databases", ICPADS 2001, 2001, pp. 26-30.

[7] Kriegel, N. and Seeger, B., "The R*-tree: An Efficient and Robust Access Method for Points and Rectangles", Proc. Of ACM SIGMOD, 1990, pp.322-331.

[8] Mutenda L., and Kitsuregawa M., "Parallel R-tree Spatial Join for a Shared-Nothing Architecture", Int'l Symposium on Database Applications for Non-Traditional Environments, 1999, pp. 429-436.

[9] Minwen Ji, "Affinity-Based Management of Main Memory Database Clusters", ACM Transactions on Internet Technology, Vol. 2, No. 4, 2002, pp.307-339.

[10] Sang K. Cha, Kihong Kim, Changbin Song, Sangyong Hwang and Yongsik Kwon, "MEADOW: a middleware for efficient access to multiple geographic databases through OpenGIS wrappers", SOFTWARE-PRACTICE AND EXPERIENCE, 32, 2002, pp.377-402.

[11] Shashi Shekhar, Sivakumar Ravada, Vipin Kumar, Douglas Chubb and Turner, "Declustering and Load-Balancing Methods for Parallelizing Geographic Information Systems", IEEE Transactions on Knowledge and Data Engineering, Vol. 10, No. 4, July 1998, pp.632-655.

[12] Zu-Kuan Wee, Jae-Dong Lee, Hae-Young Bae, "Efficient Spatial Transmission in Web-Based GIS", Workshop on Web Information and Data Management, 1999, pp.38-42.

[13] 김진덕, 서영덕, 홍봉희, "고정 그리드를 이용한 병렬 공간 조인의 태스크 할당에 관한 연구", 정보처리학회논문지D, 제8-D권 4호, 2001.

[14] 고주일, 이환재, 배해영, "분산 공간 DBMSD에서의 효율적인 공간 릴레이션 분할 기법을 이용한 병렬 공간 조인 기법," 개방형지리정보시스템학회 논문지, 제4권 제1호, 2002, pp.39-46.

[15] Schwtman H., "CSIM User Guide: CSIM 19 Simulation Engine(C, C++ Version)", Mesquite Software, Inc., 2001.

[16] K. Watkins, "Discrete event simulation in C", McGraw-Hill, 1993.

이재동

1985년 인하대학교 전자계산학 졸업 (학사)

1991년 Cleveland State University-Computer & Information Science 졸업(석사)

1996년 Kent State University-Computer Science졸업(박사)

1996. 9~1997. 3 (주)두루넷(기술기획팀)

1997. 3~현재 단국대학교 정보컴퓨터학부 컴퓨터과학전공 교수

관심분야 (Mobile) Internet Technologies/ Applications, High Performance Computing (Clustering systems etc.), GIS Technologies and Applications, Many aspects of parallel/distributed processing